无线网络干扰最小化问题的一种缩边贪心算法 *

杜　磊[1]，付喜梅[2]，杨文瀚[1]

(1. 华南师范大学 计算机学院，广州 510631; 2. 韶关学院 数学与统计学院，广东 韶关 512005)

摘　要：无线传感器网络的最大干扰最小化问题可以被描述为已知平面上 n 个点的位置及发射半径的阈值，要求设置发射半径使得任意点被其他点发射范围覆盖的最大数量达到最小。为了有效求解该问题，提出了一种新的贪心算法——缩边算法。不同于已有算法的构图方式，该算法是通过采取缩边的方式，构造网络通信拓扑图，并结合了操作系统中批量处理的思想对贪心算法进行了加速，缩短了算法的运行时间。通过实验验证，该算法相比于已有算法在随机产生的算例上能产生更优的解。

关键词：无线传感网络；贪心算法；干扰最小化；批量处理；连通图；缩边策略

中图分类号：TP301　　doi: 10.19734/j.issn.1001-3695.2018.10.0804

Edge shrinking greedy algorithm for minimizing interference in wireless networks

Du Lei[1], Fu Ximei[2], Yang Wenhan[1]

(1. *School of Computer*, *South China Normal University*, *Guangzhou* 510631, *China*; 2. *School of Mathematics & Statistics*, *Shaoguan University*, *Shaoguan Guangdong* 512005, *China*)

Abstract: The problem of minimizing the maximum interference in wireless sensor networks can be described as follows: given n points with their positions on the plane and thresholds of transmission ranges , for each point, the maximum number of other points whose transmission ranges cover this point is minimized. To effectively solve this problem, this paper proposed a new greedy algorithm called edge shrinking algorithm. The algorithm constructs a desirable network communication topology by shrinking edges, which is different from existing algorithms and is accelerated by incorporating the idea of batch processing in operating system to shorten its running time. The result shows that this algorithm can produce better solutions than existing algorithms.

Key words: wireless sensor network; greedy algorithm; interference minimization; batch processing; connected graph; shrinking strategy

0　引言

无线传感器网络（wireless sensor network, WSN）是由大量的静止或移动的传感器以自组织和多跳的方式构成的无线网络，以协作地感知、采集、处理和传输网络覆盖地理区域内被感知对象的信息，并最终把信息发送给网络的所有者。无线传感器网络具有的众多类型的传感器，可探测包括地震、电磁、温度、湿度、噪声、光强度、压力、土壤成分、移动物体的大小、速度和方向等周边环境中多种多样的现象[1,2]。

无线传感器网络一般部署在人们不方便直接控制的地方，而且传感器节点一般使用的是电池，很难重新充电或者更换电池，因此能源成为制约无线网络服务时限的关键因素之一[3]。节省节点能源的一种方法是减少附近节点同时传输信号所产生的干扰。

目前减少干扰的方法有频率分配多路技术、时间分配多路技术和拓扑控制三种。前面两种方法虽然都可以有效地减少干扰，但是要求较高的技术和成本，而拓扑控制是通过给每个节点分配一个合理的发射半径并形成一个连通的拓扑，并且减小半径的某个目标函数使网络的最大干扰最小。因此，在无线传感器网络中，拓扑控制成为了减少干扰的重要方法之一[4]，也是近年来的一个研究热点。

在平面上计算无线传感器网络的最大干扰最小化并保持网络的连通已经被证明是 NP 难度的[5]，因此要在多项式时间内找到精确的解是困难的。文献中求解该问题的算法并不多，典型的有如下几种：2008 年，Halldorsson 提出的结合了计算几何和 ε-网络理论的近似算法[6]，其最大干扰是$\Delta^{0.5}$，其中Δ是发射半径相同的网络中节点最大度。同年又有学者证明了任何能够产生连通图的传输半径函数都可以修改为具有以下特点的半径函数,其产生的拓扑结构干扰基本不变，且所有传感器节点的发射半径小于 R_{min}，R_{min} 为保持网络强连通的最小可能半径[7]。2009 年，又有人提出基于每个节点与最近邻居关系的最近邻算法[8]。2010，Aslanyan 等人提出了一种基于部分成员集合覆盖的近似算法[9]。2014 年，有人基于 Prim 算法提出了一种时间复杂度为 $O(n^2)$的最佳邻算法[10]。之后又有人尝试在最佳邻算加入了后悔机制[11,12]，在适当增加时间的前提下，一定程度上提高了算法的精确度。

本文考虑了用拓扑控制方法来减少平面中无线传感器网络中最大干扰的最小值，提出了一种不同于已有算法构图方式的新的贪心算法，并在大量的随机算例上验证其有效性。

1　问题模型以及相关算法

针对无线通信中减少干扰的问题现已提出多个不同的模

收稿日期：2018-10-18；**修回日期：**2018-12-10　**基金项目：**国家自然科学基金资助项目（61370003）；韶关学院科研项目（S201501019）

作者简介：杜磊（1994-），男，广东清远人，硕士研究生，主要研究方向为算法设计与分析（531251254@qq.com）；付喜梅（1979-），女，江西高安人，讲师，硕士，主要研究方向为计算机支持协同设计技术、 算法设计与分析；杨文瀚（1990-），男，安徽安庆人，硕士研究生，主要研究方向为算法设计与分析.

型[13]。其中有一种是以发送者为中心的模型，在此模型中干扰指的是边的干扰。另一种以接收者为中心的模型中，干扰指的是点的干扰，任意一点 v 的干扰定义为传输范围可覆盖 v 的其他节点的数量。以发送者为中心的模型只考虑了发送者但实际干扰发生在接受者上，模型不能反映真实的情况。在本文中采用以接收者为中心的模型，通过拓扑控制来设计以接受者为中心模型的无线传感器网络中干扰最小化问题的有效算法。

本文中将使用图论的知识来描述问题模型，因此需要了解一些相关的概念。由于将无线传感器网络建模为图，所以本文中的“图”和“网络”是通用的。关于图的相关概念可以参看相关文献[14]。下面给出相关的概念，然后给出形式化的定义并介绍最小干扰化模型。

平面中的无线传感器网络可以描述为节点的集合，在集合中的每个点 i 都有一个二维坐标($x(i)$，$y(i)$)，并且满足其半径函数 $r(i)\in[0,R(i)]$，其中 $R(i)$是该节点的半径阈值。因此，对于传输半径函数 r 所导出的所有节点的传输关系可以建模为一个图 G_r=<V,E_r>。对于 V 中的任意两点 u 和 v，边(u,v)存在当且仅当 $r(u)$和 $r(v)$不小于 u 和 v 的欧氏距离 $d(u,v)$，即 $r(u)\ge d(u,v)$且 $r(v)\ge d(u,v)$。本文为了方便起见，将 G_r=<V,E_r>叫做传输半径函数的导出图，并作出以下定义。

定义 1　对 V 中任意的两点 $u,v\in V$，经由传输半径函数 r 导出的任意点之间的传输关系记为图 G_r=<V,E_r>，边$(u,v)\in E_r$ 存在当且仅当 u 和 v 的传输半径 $r(u)$和 $r(v)$不小于它们之间的欧氏距离 $d(u,v)$，即 $r(u)\ge d(u,v)$且 $r(v)\ge d(u,v)$。因此，G_r=<V,E_r>称为传输半径函数 r 的导出图。

定义 2　导出图 G_r=<V，E_r>中任一点 v 的干扰度定义为 $I_r(v)$，表示发射半径能覆盖到节点 v 的图中其点的个数，即

$$I_r(v)=|\{u\in V\setminus\{v\}\,|\,r(u)\ge d(u,v)\}| \quad (1)$$

定义 3　导出图 G_r=<V,E_r>的干扰度 I_r 定义为图中所有点干扰度的最大值，即

$$I_r=\max\{I_r(v)\,|\,v\in V\} \quad (2)$$

根据定义 2 和 3，图 1 中点 v 的干扰是 3，导出图的干扰也是 3。

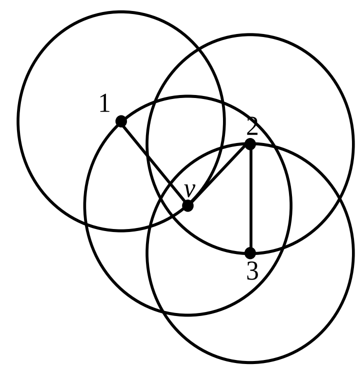


图 1　最大干扰示意图

Fig. 1　Example diagram of maximum interference

1.1　干扰最小化模型

考虑到连通的最基本的要求，可以有下面的问题模型

最小干扰化问题（MIP）:给定平面中的一个节点集 V，V 中的任意一点 i 的坐标可表示为($x(i)$，$y(i)$)。对 V 中的任意一点 i 都有一个满足以下条件的传输半径阈值函数 $R(i)\in[0,1]$ 且导出图 G_R=<V,E_R>连通。因此需要确定一个传输半径 r 满足导出图 G_r=<V,E_r>是连通的且干扰最小的。

从上文的定义可知，对任意一个传输半径 r 都满足导出图 G_r 是导出图 G_R 的子图。而且，如果重新选择使节点 v 到它在图 G_r 中距离最远的邻居的距离作为节点 v 的传输半径，最终的得到的导出图和导出图 G_r 有相同的干扰。因此，可以仅从节点 v 与其在导出图 G_R 中可到达的邻居节点的不同距离中来选择 v 的传输半径。通过这样的选择方式，节点 v 的传输半径即导出图 G_r 中 v 与可到达的距离最远的邻居的距离。

MIP 已经被证明是 NP 难度的，即使对于寻找所有节点都在一条直线上这种特殊的 MIP 的一个最优实例也是很困难的，尽管难度还没有被证明出来。对于二维网络的特殊情况，如网格网络，这个问题的难度也是 NP 难度的。MIP 难度源于其组合结构，以至于从许多直观的特性，如最小度、最近的邻居、最好的邻居等特点，都不能产生令人满意的解。因此可以考虑设计贪心算法来解决这个问题。

1.2　干扰最小化问题的典型算法

目前已有不少学者研究了干扰最小化问题，并提出了解决该问题的相关算法，本文主要介绍了介绍如下几种算法。

1.2.1 最近邻算法(NNA)

贪心算法的基本思路是从问题的某一个初始解出发一步一步地进行，根据某个优化测度，每一步都要确保能获得局部最优解。在干扰最小化问题中，可直观地认为节点的传输半径越小则网络中的干扰越小。最近邻算法正是基于这样的直观特性，并且结合了最小生成树的思想，通过调整两个相邻最近的节点的传输半径来连通当前网络中不连通的两个连通分量。重复这个步骤，直至整个网络连通。

1.2.2 最佳邻算法(BNA)

最近邻算法在一般网络中的效果尚可，但是在特殊网络（指数链网络）中效果很差。针对最近邻算法的不足，文献[10]提出了改进的最近邻算法——最佳邻算法。最佳邻算法的主要思想是：通过调整当前连通网络最大干扰增幅最小的节点对的传输半径来连通两个不连通的部分。重复上述步骤，直至整个网络连通。

最佳邻算法与最近邻算法的时间复杂度相当。但是大量的实验结果表明，最佳邻算法不仅能消除最近邻算法在特殊网络上的弊端，在普通网络上的效果也优于最近邻算法。

1.2.3 后悔贪心算法(RGA)

Sun[12]在 2017 年提出了后悔贪心算法，其主要步骤遵循贪心算法的工作流程，并在贪心算法的每个迭代步尝试采用后悔策略。在贪心算法的每个迭代步中，如果当前网络的最大干扰发生了改变，即当前导出图的最大干扰增加后，将针对此时具有最大干扰的节点用后悔策略调整某些节点的已由贪心算法确定的传输半径来试图降低它的干扰，使得当前最大干扰减少。

1.2.4 旋转贪心算法(CGA)

基于基本贪心算法 G 的旋转贪心算法的主要思想是在贪心算法 G 求得的可行解中选择一定比例的元素作为部分解。每次将部分解的第一个元素移除，通过贪心算法 G 重新加入元素，更新部分解。重复这一步骤，直至迭代结束。最后利用贪心算法 G 得到一个完整的解。

1.2.5 基于部分成员集合覆盖算法(SCA)

Kuhn 等人在 2005 提出了最小成员集合覆盖问题，并将其用于解决蜂窝网络中的干扰问题。在文献[9]中，Aslanyan 等人在 2010 年将最小成员集合覆盖问题推广到最小部分成员集合覆盖，将其用于求解无线网络干扰最小化问题，并从理论上证明了干扰的上限 $O(opt^2\times ln^2n)$和近似比 $O(opt\times ln^2n)$。

部分成员集合覆盖定义如下：给定一个 n 元素集合 $S=S_1\cup S_2$ 以及 S 子集的集合 $C=\{C_1, C_2, \ldots C_k\}$,其中 $S_1\cap S_2=\phi$,$|S_1|=n_1$, $|S_2|=n_2$，求集合 $C'\subset C$ 且满足以下两个条件：

C'覆盖 S_1 中的所有元素，可表示为 S1 $S_1 \subseteq \cup_{C_i \in C'} C_i$ 。

C'使 S_2 中的元素被覆盖的最大次数最小，即 $\max_{u \in S_2} M(u,C')$ 最小。其中 S_2 中的元素 u 被集合 C'覆盖的次数定义为元素 u 的成员值 $M(u,C')$，可表示为 $M=(u,C')=|C_i \in C':u \in C_i|$。

然后将该问题构建成一个整数规划问题，利用线性规划松弛技术解决，因此可以将 C'看做是问题的一个解，对于每一个 $C_j \in C$，分配一个 $x_j \in \{0,1\}$，即 $x_j=1 \Leftrightarrow C_j \in C$。

Minimize　z

to
$$\sum_{u_i \in C_j} x_j \geq 1 \qquad u_i \in S_1$$
$$\sum_{u_i \in C_j} x_j \leq z \qquad u_i \in S_2$$
$$x_j \in \{0,1\} \qquad C_j \in C$$

具体步骤如下：

a)初始化图 $G=<V，E_0>$，$E_0=\phi$，$l=0$。

b)$l=l+1$，找到问题中对应的 S_1、S_2、C，并用线性规划求 C'

c)调整 C'中边上的半径，$E_l=E_{l-1} \cup C'$，更新图 $G_l=<V,E_l>$。

d)重复步骤 b)和 c)，直至图连通为止。

2　缩边贪心算法(SEA)

2.1　基本思想

缩边贪心算法（简称缩边算法，以下均使用简称）主要思想是从最大传输半径 R 的导出连通图 $G_R=<V,E_R>$开始，通过逐步缩小图中各点的传输半径，找到干扰较小的连通子图 $G_r=<V,E_r>$。算法开始时通过将所有节点的传输半径初始化到最大的阈值，即 $r(i)=R[i]$，$i \in V$，即初始化的导出图是连通的，在确保导出图是连通的前提下算法按照缩边策略去减小各节点的发射半径，直至所有节点的传输半径都不能再减小为止。而已有算法都是通过初始化图中各点发射半径为 0 开始，然后一步一步地按照算法中的策略扩大某些节点的发射半径使得这些点加入到当前导出图的连通分图中，即通过“加边”的方式将这些节点连通到当前的连通图中，与之相比，缩边算法又可以看做是不断地对当前导出连通图通过“减边”的方式以达到降低干扰的目的。

由于最终所得的导出图必须是连通的，所以选点缩边要遵循的首要条件是不能破坏导出图的连通性。根据这条件选择缩边的节点，直观上所选的点应是出度大且最大干扰点应在其覆盖圆的最外围。出度大意味着该点的连通性能可能较好，缩小其发射半径对导出图的连通性影响不大；而最大干扰点在其覆盖圆最外围的点则意味着小幅度的缩小点的发射半径即可消除对最大干扰点的干扰，最小程度地降低了对导出图连通性的影响。然而“出度大”和“最大干扰点处在其外围”这两点如果仅仅单一考虑一个都会出现极端的情况，会极大地影响当前导出图的连通性。为了避免出现这些极端情况，下面综合考虑这两点来选取缩边的节点。为此作出如下定义：

设 u 为当前最大干扰点，令 $N(u)$为对 u 产生干扰的节点集合，即 $N(u)=\{v \in V|r_v \geq d(u,v)\}$,令 $l(u,v)$为 u 在 v 中按与 v 距离从近到远的排序号，其中 $v \in N(u)$，则选择缩边的点为

$$v' = \arg\max_{v \in N(u)} l(u,v) \tag{3}$$

缩边点的发射半径由 $r_{v'}$缩小为 $d(u,v')-\varepsilon$，其中 ε 为任意小的正实数。

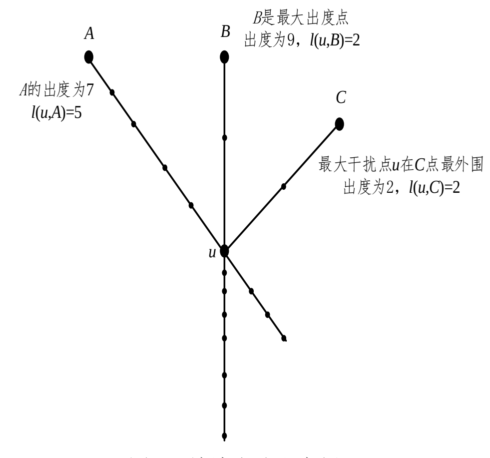


图 2　缩边点选取实例

Fig. 2　Example of selecting shrinking point

图 2 中直观展示了按式(3)选取缩边点的合理性，假设 u 点为当前最大干扰点且最大干扰为 3（为了方便理解，将所有在传输半径以内的点都放在同一直线上），点 A、B、C 分别为干扰 u 的三个节点，可看出 B 的出度最大，而 u 处于 C 的发射范围的最外围，当缩小节点 B 或节点 C 的发射半径都会使得缩边的节点仅与一个节点相连，均会破坏导出的连通性。根据式(3)选取的点为 A，缩小点 A 的发射半径能保持导出图的连通性。而如果选择点 B 或者 C 都会极大地破坏导出图的连通性。

2.2　算法描述

在本节中将对缩边算法进行描述分析，并根据各种算法在相同实例上的执行结果来比较不同算法产生的解的质量。算法 1 给出缩边算法的伪代码。

算法 1　缩边算法

```
Input:已知节点集 V={1,2,...,n}在平面上的坐标，对于每个点 i∈V 的最大传输半径阈值函数 Rᵢ ∈[0,1],满足图 G_R=(V,E_R)是连通的。
Output: 图的传输半径向量 r = (r₁,r₂,...,rₙ)，使得图 G_r=(V,E_r)连通且干扰较小。
Begin
1. rᵢ=Rᵢ, i∈{1,2,...,n}
2. do  u←当前最大干扰度点
3. vᵢ∈V, N(u)←{v∈V|r_v≥d(u,v)}
4. While N(u)≠Φ
     v' = arg max_{v∈N(u)} l(u,v)
5. if 对 v'缩边后图仍连通 then
6. do  更新 r_v'
7. else
8. N(u)←N(u)\{v'}
9. end if
10. end while
11. while  u 的干扰减小
12. end while
end
```

下述定理给出了缩边算法的正确性及时间复杂度的结论。

定理 1　该算法在 $O(n^3 \times d)$时间内产生一个传输半径函数 r 且导出图 $G_r=<V,E_r>$是连通的。

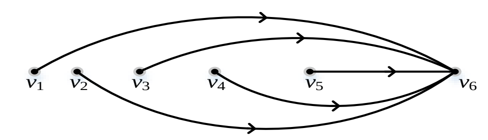
证明　首先证明算法的时间复杂度。该算法第一步首先要找到当前网络中最大干扰节点 u，需要时间 $O(n)$；接着第

二步要找出所有出度中含有 u 的节点需要时间 $O(n)$；最后为了保证图的连通性，在每次缩小节点的传输半径之后需要判断新的导出子图是否连通需要时间 $O(n)$。又因为该算法需要重复上述直至再也找不到可以缩小的节点为止，一般这个重复次数 d 会远小于 n，所以该算法的时间复杂度为 $O(n^3 \times d)$。

其次证明导出图的连通性，该算法初始化图时，就确保了图是连通的，并且每一步选择缩小传输半径的节点都必须以不能破坏图的连通性为首要条件，因此所选择的点不影响图的连通性，所以该算法产生的导出图 $G_r=<V,E_r>$是连通的。证毕。

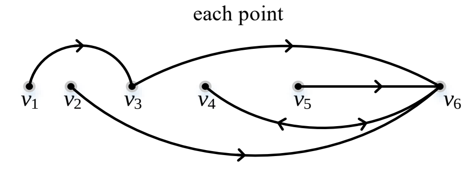
2.3　算例分析

例 1　对于上述文中所提到的一维指数链网络的实例，考虑在一维指数链网络 n=6 的实例上用缩边算法的执行情况。初始化图时，所有的节点都连接到了与自己相距最远的节点，如图 3(a)所示，其中每个点的弧线表示该节点的传输半径最远能影响到的节点。



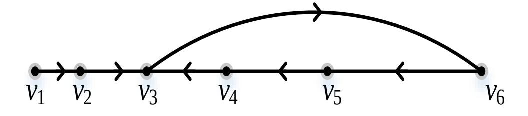
(a)在 6 个点的一维指数链网络,缩边算法初始化各点半径产生的导出图

(a) In one-dimensional exponential chain network with six points, edge reduction algorithm initializes derived graphs generated by radius of each point



(b) 执行一次算法迭代步后的导出图

(b) Derived graphs after performing iteration step of algorithm



(c) 在 6 个点的一维指数链网络，缩边算法的最终结果导出图

(c) In one-dimensional exponential chain network with six points, final result of edge reduction algorithm is derived

图 3　缩边算法在一维指数链实例

Fig. 3　Example of shrinking algorithm in one-dimensional chain

此时图中 6 个节点的干扰度都为 5，均为最大干扰点。最大干扰节点不止一个时，按顺序先队编号小的点先进行缩边，选取的需要减少干扰的节点为 v_1，则缩边节点 $v' = \arg\max_{v \in N(v_1)} I(v_1, v) = v_6$。依此类推，接下来需要减少干扰的节点分别为 v_2、v_3、v_4 对应的缩边节点分别为 v_6、v_6 和 v_1，完成这些缩边操作之后，各节点传输半径如图 3(b)所示。经过上述一系列缩边之后，节点 v_1 和节点 v_6 都缩小了传输半径，使得当前图的最大干扰为 4。

重复上述步骤，直至不存在可以缩边的节点为止，最终结果示意图如图 3(c)所示。图 3(c)中，v_1、v_2、v_4、v_5 和 v_6 的传输半径都仅仅覆盖到了与自身相邻最近的两个节点（除端节点外），而节点 v_3 的传输半径则覆盖到了与自身相邻最远的一个节点即 v_6。从示意图中可直观地观察到导出图是连通的，且最大干扰度为 2，平均干扰度为 1.83。

例 2　考虑在二维空间随机产生了 n=10 个点的算例上用缩边算法的执行情况并与已有的算法作了对比分析。算例中 10 个点 $v_1,v_2,\ldots,v_{10}$ 在平面上的位置如图 4 所示，对应每个点的坐标分别为(1.11，0.17)(1.64，1.84)(1.92，1.72)(0.46，0.79)(1.94，0.81)(0.29，1.84)(0.24，1.30)(0.60，1.00)(1.56，0.37)(1.54，2.26)。

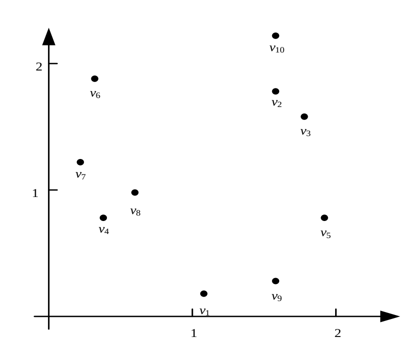


图 4　节点在平面上的位置

Fig. 4　Position of points on plane

已知上述平面坐标的节点集 V={1,2,...,10}，算例的最大传输半径阈值函数，满足图 $G_R=(V,E_R)$是连通的。缩边算法执行时进行初始化，即将每个节点的传输半径设置为所能连接到最远的节点。初始化示意图如图 5 所示，图中箭头表示该节点所能到达最远的节点。

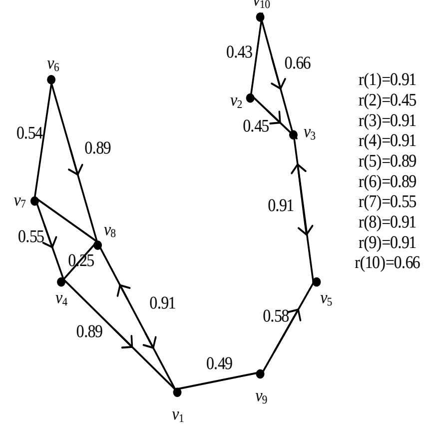


图 5　各节点传输半径初始化

Fig. 5　Initialize transmission ranges of points

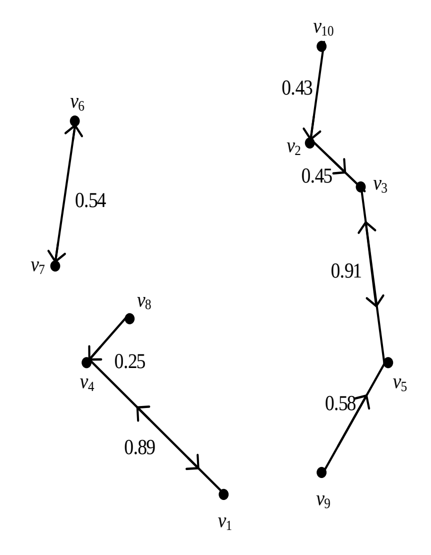


图 6　最终各点传输半径

Fig. 6　Final transmission ranges of points

缩边算法每个迭代步中节点传输半径变化如下：

a)$r(1)$=0.91→0.89
b)$r(8)$=0.91→0.89
c)$r(10)$=0.66→0.43
d)$r(7)$=0.55→0.54
e)$r(8)$=0.89→0.25
f)$r(6)$=0.89→0.54

最终结果示意图见图 6。为了清晰起见，这里仅标记出了经过一系列缩小传输半径后各个节点传输半径内所最远能到达的节点。从图中各节点的传输半径可知，导出图是连通的，最大干扰点是有多个分别是 v_1、v_2、v_3、v_4、v_5、v_7、v_8、v_9，其干扰值均为 2。

在该算例中前文提到的“加边”算法在这个算例中都得出了相同的结果，其中最大干扰为 3，如图 7 所示。

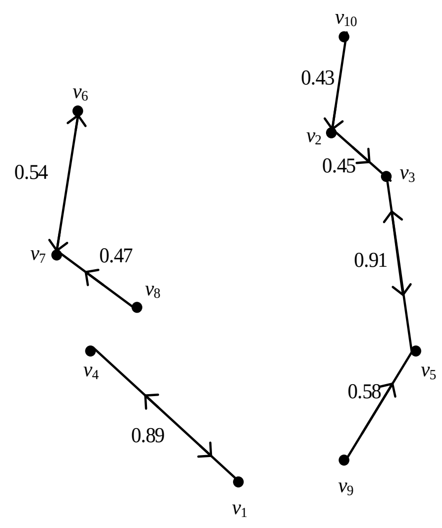


图 7　加边”算法最终结果

Fig. 7　Result of adding edge algorithm

将图 6 与 7 对比可看出，图 6 中唯一的区别在于缩边算法中设置 $r(8)$=0.25，而“加边”算法中设置 $r(8)$=0.47，从而导致最终最大干扰值不同。因此在这个实例上缩边算法所得的解比“加边”算法得到的解更优。

2.4　实验结果

实验环境如表 1 所示。

表 1　实验环境说明

Table 1　Experimental environment description

Type	Specific information
OS	Microsoft Windows 7 Pro version 64bit
CPU	Intel(R) Core(TM) i5-2410M CPU @ 2.30GHz
Memory	6GB
Disk	HHD 512GB
IDE	Codeblocks 13.12

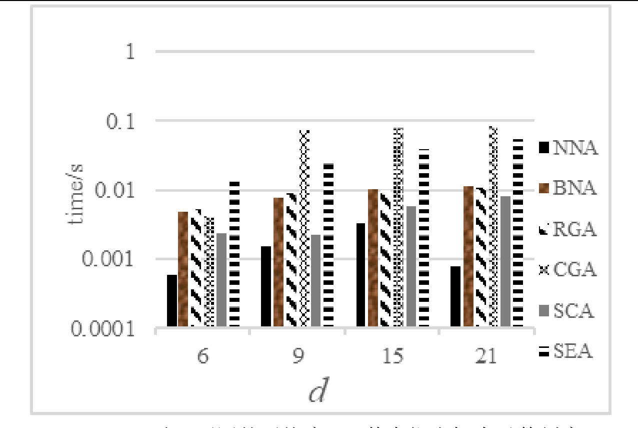
在这一部分，通过比较缩边算法与其他算法在相同例子上的执行结果来对比不同算法之间产生的解的效果。

每个例子产生规则如下：n 代表实例中点的个数，R 是每个点的传输半径的阈值（一般为 1），d 是到处图 $G_R=(V,E_R)$ 的平均点度，可表示导出图中边的稠密度。对于每个给定的点对(n,d)，在 $w\times w$ 的方形区域内随机产生 n 个点，容易知道 d 满足 $d=n\pi R^2/w^2$。对每个实例，比较每个算法产生的导出图的干扰度。随机产生 100 个实例并用这 100 个实例的干扰度的平均值作为点对(n,d)的最终干扰。根据下列的图 8、表 2 可以直观地看出各算法的解的效果，以及运行时间的对比。如表 2 所示，其中算例(n,d)中 n 和 d 分别代表相应例子的节点数和平均点度

表 2　各个算法所得解的平均目标值

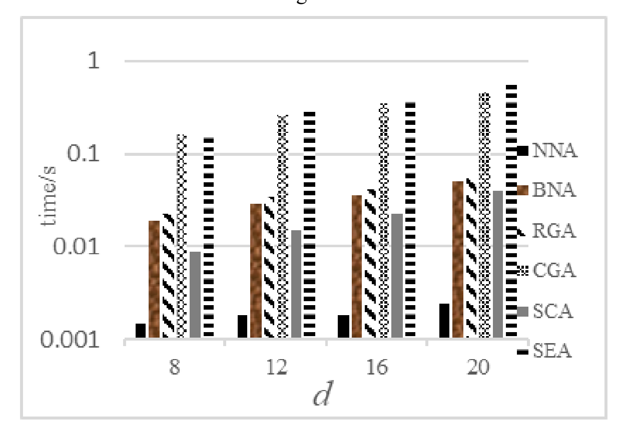
Table 2　Average result of algorithms

(n,d)	NNA	BNA	RGA	CGA	SCA	SEA
(100,6)	5.01	4.62	4.40	4.41	4.74	4.02
(100,9)	5.37	4.56	4.33	4.25	5.13	4.01
(100,15)	5.29	4.57	4.16	4.08	5.08	3.95
(100,21)	5.08	4.47	4.09	3.98	4.88	3.81
(200,8)	5.57	4.79	4.55	4.39	5.20	4.06
(200,12)	5.43	4.66	4.43	4.35	5.13	4.03
(200,16)	5.50	4.68	4.33	4.20	5.20	4.03
(200,20)	5.57	4.70	4.36	4.25	5.25	4.10



(a) n=100 时，不同的平均度 d，其中纵坐标为对数刻度

(a) When n = 100, the average degree D is different, in which ordinates are logarithmic



(b) n=200 时，不同的平均度 d，其中纵坐标为对数刻度

(b) When n = 200, average degree D is different, in which ordinates are logarithmic

图 8　各算法在不同随机图上平均运行时间

Fig. 8　Average running time of algorithms in random graphs

从表 2 可以看出，缩边算法的解的效果明显优于其他算法，但是在图 8 中可以直观地看出缩边算法的运行时间比其他算法长很多，并且随着点对(n,d)的增长，运行时间呈指数增长，这说明单纯的缩边算法适用性并不太强。为此本文结合了增强贪心算法的策略，对缩边算法进行加速。

3　缩边算法的加速

3.1　加速策略

增强型贪心算法的策略有很多，如迭代贪心算法已应用于置换流水车间调度问题，Xiong 等人将随机贪心算法应用于 MLST 问题[15,16]。这两类贪心算法易于实现，但在 MIP 问题中随机化容易破坏图的连通性。最近 Carmine 等人提出了旋转贪心算法，易于实现且已成功用于增强一些典型问题的

贪心算法[17]。本文的第二章对旋转贪心算法进行了解释，并实验部分进行相关了的实验对比。

本文考虑了结合批量处理的增强策略，对缩边算法进行了改进。批量处理增强策略就是多个步骤同时运行，与计算机上批量处理的基本思想相似，这种策略可以减少算法的运行时间，使算法的适用性更广。

对于影响缩边算法运行时间的因素主要分为四个部分：a)找到当前最大干扰点；b)找出所有出度中含有当前最大干扰点的节点；c)缩边并判断图的连通性；d)循环上述步骤，直至不存在可缩边的节点。

关于 a)，由于当前连通图的最大干扰节点在经过一次缩边策略之后，会有全局性的改变，所以每次搜索都需要遍历全图。关于 b)，可以采取加入索引的方式，但是索引需要用到矩阵存储，空间复杂度会增加 $O(n^2)$，并且在每次对节点进行缩边之后需要对矩阵进行修改，难以对该部分进行优化。因此将对 c)和 d)进行优化，尽可能减少图的连通性的判断次数，并加快图的最大干扰收敛的速度。

从上述算法描述中可以知道判断图的连通性是在缩小一个节点的传输半径之后进行。通过大量实验发现，最大干扰度点不仅仅只有一个，即需要减少干扰度的节点可以存在多个，但是在缩边算法中只能依次去对这些同样具有当前最大干扰度点去进行缩边策略，因此每进行一次缩边策略就需要判断一次图的连通性，这样就会导致运行时间过长。对此本文考虑是否可以将计算机批量处理的思想结合到缩边算法中，通过所有具有当前最大干扰度点一次性的进行缩边策略之后再进行判断导出图的连通性。具体算法描述如算法 2 伪代码。

算法 2：批量缩边算法

Input：已知节点集 $V=\{1,2,...,n\}$在平面上的坐标，对于每个点 $i\in V$ 的最大传输半径阈值函数 $R_i\in[0,1]$，满足图 $G_R=(V,E_R)$是连通的。

Output：图的传输半径向量 $r=(r_1,r_2,...,r_n)$，使得图 $G_r=(V,E_r)$连通且干扰较小。

```
Begin
r_i=R_i, i∈1,2...n, V_I_max=Φ
do  I_max←当前导出图最大干扰
for  each  v∈V  Do
if  I_v=I_max  then
do V_I_max← V_I_max∪v
end if
end for
if  V_I_max 中的点都进行缩边操作后图仍连通 then
do 更新各个缩边点的半径
else
对 u∈V_I_max 采取未改进的缩边算法
end if
while  I_max 减小
end while
end
```

3.2 加速后的实验结果

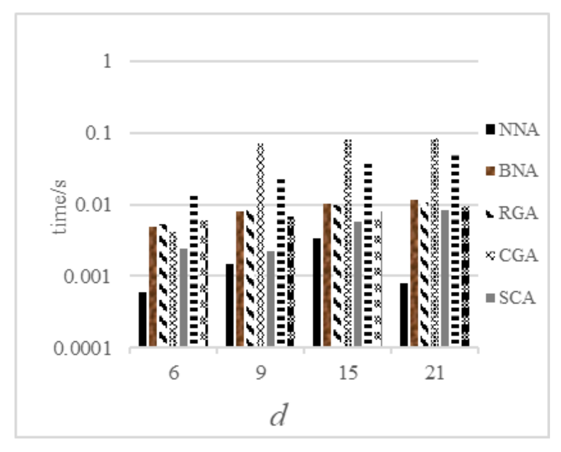
为了说明算法的改进是有效的，本文对批量缩边算法进行了对比实验，实验数据如表 3 和图 9 所示。其中算例(n，d)中 n 和 d 分别代表相应例子的节点数和平均点度。

根据上述图表可以看出，虽然批量缩边算法的解的效果与原缩边算法的效果对比略逊一筹，但是相差并不大，还是优于其他算法。反观批量缩边算法的运行时间，批量的加速策略虽不能将缩边算法的时间复杂度从 $O(n^3)$降低一个数量级到 $O(n^2)$，但是其实际运行时间却远远低于原缩边算法的运行时间。

表 3　各算法所得解的平均目标值

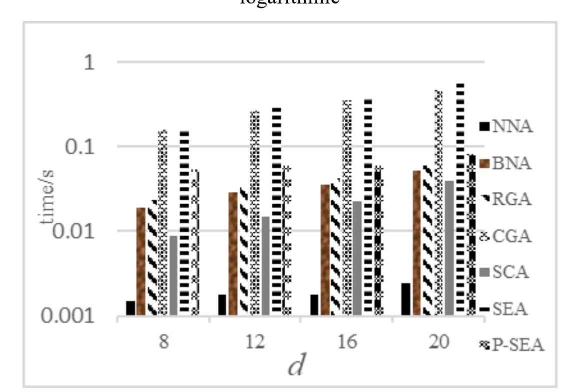
Table 3　Average result of algorithms

(n,d)	NNA	BNA	RGA	CGA	SCA	SEA	P-SEA
(100,6)	5.15	4.56	4.39	4.36	4.80	4.08	3.89
(100,9)	5.21	4.60	4.39	4.27	5.08	4.06	4.08
(100,15)	5.19	4.57	4.25	4.11	5.06	3.83	4.09
(100,21)	5.15	4.31	4.11	4.01	4.88	3.89	3.94
(200,8)	5.58	4.94	4.76	4.80	5.22	4.11	4.20
(200,12)	5.53	4.68	4.38	4.36	5.17	4.03	4.09
(200,16)	5.53	4.67	4.35	4.30	5.21	3.98	4.14
(200,20)	5.62	4.71	4.39	4.22	5.25	4.00	4.15



(a) n=100 时,不同的平均度 d,其中纵坐标为对数刻度

(a) When n = 100, average degree D is different, in which ordinates are logarithmic



(b) n=200 时,不同的平均度 d,其中纵坐标为对数刻度

(b) When n = 200, average degree D is different, in which ordinates are logarithmic

图 9　各算法在不同随机图上平均运行时间

Fig. 9　Algorithms of average running time in random graphs

4　结束语

在以接收者为中心的干扰模型下，针对无线传感器网络中基于拓扑控制技术的干扰最小化问题，本文提出了一种缩边贪心算法。该算法采用了已有算法不一样的构图方式，通过在每个节点的传输半径为最大的阈值的导出图上采取缩边的方式进行构图在大多数随机产生的算例上能产生更优解。针对其运行时间过长这一缺陷，本文将批量处理策略加入到缩边算法中，大大缩短了其运行时间，相较于其他算法使得其能在几乎相同的时间内获得更优的结果。

参考文献：

[1] 孙利民，李建中，陈渝．无线传感器网络 [M]．北京：清华大学出版社, 2005: 3-23. (Sun Li-ming, Li Jian-zhong, Chen Yu. Wireless sensor networks [M]. Beijing: Tsinghua University Press, 2005: 3-23.)

[2] Bilò D, Proietti G. On the complexity of minimizing interference in ad-hoc and sensor networks [J]. Theoretical Computer Science, 2008, 402 (1): 43-55.

[3] Akyildiz I F, Su Weilian, Sankarasubramaniam Y, *et al.* Wireless sensor networks: a survey [J]. Computer Networks, 2002, 38 (4): 393-422.

[4] Lou Tiancheng, Tan Haisheng, Wang Yuexuan, *et al.* Minimizing average interference through topology control [C]// Proc of International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics. Berlin :Springer, 2011, 7111: 115-129.

[5] Buchin K. Minimizing the maximum interference is hard [J]. arXiv: 0802. 2134v2, 2008.

[6] Von Rickenbach P, Schmid S, Wattenhofer R, *et al.* A robust interference model for wireless ad-hoc networks [C]// Proc of the 19th IEEE International Parallel and Distributed Processing Symposium. Piscataway,NJ: IEEE Press, 2005, 13 (4): 239-246.

[7] Benkert M, Gudmundsson J, Haverkort H, *et al.* Constructing minimum-interference networks [J]. Computational Geometry, 2008 40 (3): 179-194

[8] Von Rickenbach P, Wattenhofer R, Zollinger A. Algorithmic models of interference in wireless ad hoc and sensor networks [J]. IEEE/ACM Trans on Networking, 2009, 17 (1): 172-185.

[9] Halldorsson M M, Tokuyama T. Minimizing interference of a wireless ad-hoc network in a plane [J]. Theoretical Computer Science, 2008, 402 (1): 29-4

[10] Zhang Xiaoke, Chen Weidong. An effective algorithm for interference minimization in wireless sensor networks [C]// Proc of International Conference on Intelligent Computing. [S.l.]:Springer International Publishing, 2014 572-581

[11] . Hassin R, Levin A. A better-than-greedy approximation algorithm for the minimum set cover problem [J]. SIAM Journal on Computer, 2005, 35 (1): 189-200.

[12] 孙佩歆．无线传感器网络中干扰最小化问题的后悔贪心算法 [J]. 计算机工程与科学, 2017, 39 (12): 2217-2223. (Sun Peixin. A regret greedy algorithm for the interference minimization problem in wireless sensor networks [J]. Computer Engineering and Science Computer Engineering and Science, 2017, 39 (12): 2217-2223.)

[13] Shetty D P, Lakshmi M P. Algorithms for minimizing the receiver interference in a wireless sensor network [C]// Proc of IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics . 2016: 113-118.

[14] West D B. Introduction to graph theory [M]. Upper Saddle River, NJ :Prentice-Hall, 1996

[15] Karabulut K. A hybrid iterated greedy algorithm for total tardiness minimization in permutation flowshops [J]. Computers & Industrial Engineering, 2016, 98: 300-307.

[16] Xiong Yupei, Golden B, Wasil E. Improved heuristics for the minimum label spanning tree problem [J]. IEEE Trans on Evolutionary Computation, 2006, 10 (6): 700-703.

[17] Cerrone C, Cerulli R, Golden B. Carousel greedy: a generalized greedy algorithm with applications in optimization [J]. Computers & Operations Research, 2017, 85: 97-112.